

AS

T 011592991/3,AB

011592991/3,AB

DIALOG(R)File 351:Derwent WPI

(c) 2002 Derwent Info Ltd. All rts. reserv.

011592991

WPI Acc No: 1998-010119/199802

XRPX Acc No: N98-007882

Graphical programmer with keyboard, display and software tool - provides display of Nassi-Shneiderman diagram or Petri net with cursor movable by keying between gaps in arrangement of objects

Patent Assignee: SIEMENS AG (SIEI )

Inventor: ROSSKOPF H

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
DE 19621062	A1	19971127	DE 1021062	A	19960524	199802 B

Priority Applications (No Type Date): DE 1021062 A 19960524

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
DE 19621062	A1		6	G06F-003/033	

Abstract (Basic): DE 19621062 A

The programmer displays a sequential function chart with objects including transitions (2,6,7), action blocks (4,5), an OR branching (1a,1b) and an AND branching (3a,3b). The cursor (8) is moved conventionally by means of left, right, up and down shift keys of the keyboard.

The objects are arranged horizontally and vertically or in a network structure. Each operation of the cursor keys moves the cursor abruptly into the gap between two objects.

USE/ADVANTAGE - For setting-up stored control program of automation equipment. Cursor positioning within graphic display is simplified and accelerated without recourse to mouse or other equipment of insufficient electromagnetic compatibility.

?



①9 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

①2 **Offenlegungsschrift**  
①0 **DE 196 21 062 A 1**

⑤1 Int. Cl.®:  
**G 06 F 3/033**  
G 05 B 19/04

②1 Aktenzeichen: 196 21 062.3  
②2 Anmeldetag: 24. 5. 96  
④3 Offenlegungstag: 27. 11. 97

DE 196 21 062 A 1

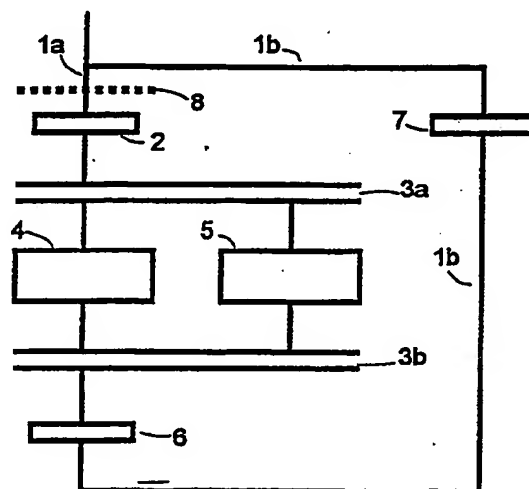
- ⑦1 Anmelder:  
Siemens AG, 80333 München, DE
- ⑦2 Erfinder:  
Roßkopf, Harald, Dipl.-Inform., 76185 Karlsruhe, DE
- ⑤6 Für die Beurteilung der Patentfähigkeit  
in Betracht zu ziehende Druckschriften:  
US 48 31 580  
Selecting Nodes in a Hierarchial Graph using the  
Keyboard. In: IBM Technical Disclosure Bulletin,  
Vol.37, No.07, July 1994, S.151-156;  
NICKLES, Michael: Kästchenstrukturierung. In:  
DOS 4/1991, S.172,173,176,177;  
TRUNZ, Wolfgang, BENDER, Klaus: Ein  
Sprachstandard zur herstellereunabhängigen  
Programmierung von SPS-Systemen. In:  
Automatisierungstechnische Praxis 33, 1991, 3,  
S.128-137;

HOLDER, Manfred: SPS-Esperanto. In: industrie-  
elektrik + elektronik, 33.Jg., 1988, Nr.1, S.16-18;  
Cursor Control Method For Formula Editing. In: IBM  
Technical Disclosure Bulletin, Vol.31, No.2, July 1988,  
S.241-243;

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤4 Programmiergerät

- ⑤7 Es ist ein Programmiergerät bekannt, welches eine Tasta-  
tur und eine Anzeigeeinheit sowie ein Software-Werkzeug  
zum Erstellen eines Steuerprogramms für ein Automatisie-  
rungsgerät in Form einer Grafik umfaßt, die mit auf der  
Anzeigeeinheit darstellbaren Objekten versehen ist, wobei  
die Objekte horizontal und vertikal oder in Form einer  
Netzstruktur angeordnet sind und zueinander in Beziehung  
stehen. Um die Cursorpositionierung innerhalb der Grafik zu  
vereinfachen, ist vorgesehen, daß in der Grafik der Cursor  
anzeigbar ist und durch Betätigen einer Cursortaste die  
Cursorpositionierung objektweise erfolgt.  
Die Erfindung wird angewandt in Programmiergeräten für  
speicherprogrammierbare Steuerungen.



DE 196 21 062 A 1

Die Erfindung betrifft ein Programmiergerät, welches eine Tastatur und eine Anzeigeeinheit sowie ein Software-Werkzeug zum Erstellen eines Steuerprogramms für ein Automatisierungsgerät in Form einer Grafik umfaßt, die mit auf der Anzeigeeinheit darstellbaren Objekten versehen ist, wobei die Objekte horizontal und vertikal oder in Form einer Netzstruktur angeordnet sind und zueinander in Beziehung stehen.

Ein derartiges Programmiergerät ist aus dem Siemens-Katalog PLT 150, Ausgabe 1996, Kapitel 5 bekannt. Um z. B. einen Aktionsblock an einer Stelle innerhalb einer Funktionsplangrafik (Sequential Function Chart (SFC)), z. B. zwischen zwei Objekten, einzufügen, wählt ein Programmierer zunächst aus einem vorgegebenen Bereich auf der Anzeigeeinheit den Aktionsblock mit Hilfe einer Bedieneinheit in Form einer "Maus" aus, positioniert den Mauscursor auf die Verbindungslinie der beiden Objekte und fügt schließlich den ausgewählten Aktionsblock an dieser positionierten Stelle ein, indem er eine Maustaste betätigt. Der Mauscursor ist dabei allerdings weitgehend genau auf die Verbindungslinie zu positionieren, ansonsten ist die Positionierung fehlerhaft, und die gewünschte Stelle muß erneut mit dem Mauscursor angewählt werden, um den Aktionsblock einfügen zu können.

Der vorliegenden Erfindung liegt die Aufgabe zugrunde, ein Programmiergerät der eingangs genannten Art zu schaffen, welches die Cursorpositionierung innerhalb der Grafik vereinfacht.

Diese Aufgabe wird dadurch gelöst, daß in der Grafik ein Cursor anzeigbar ist und durch Betätigen einer Cursortaste die Cursorpositionierung objektweise erfolgt.

Durch die objektweise Positionierung des Cursors ist schnell die gewünschte Stelle innerhalb der Grafik anwählbar, wobei die Grafik eine Funktionsplangrafik, ein Nassi-Shneiderman-Diagramm, ein Petri-Netz oder eine andere geeignete grafische Darstellung eines Steuerprogramms sein kann.

Die Erfindung ist insbesondere dort einsetzbar, wo hohe EMV-Anforderungen an elektronische Geräte gestellt werden. Es kann auf die "Maus" als Bedieneingabe verzichtet werden, die gewöhnlich diesen EMV-Anforderungen nicht genügt.

In einer Ausgestaltung der Erfindung sind Objekte in Gruppen zusammenfaßbar, wobei die Cursorpositionierung objektgruppenweise erfolgt. Dadurch wird eine schnelle Cursorpositionierung auf der Anzeigeeinheit ermöglicht.

Weitere Ausgestaltungen der Erfindung ergeben sich aus den weiteren Unteransprüchen.

Anhand der Zeichnung, in der ein Ausführungsbeispiel der Erfindung veranschaulicht ist, werden im folgenden die Erfindung sowie Ergänzungen und Weiterbildungen näher beschrieben und erläutert.

Hierzu zeigen die

Fig. 1 bis 3 auf einer Anzeigeeinheit eines Programmiergerätes darstellbare Funktionsplangrafiken (Sequential Function Charts).

In Fig. 1 sind mit 1 ... 7 Objekte bezeichnet, von denen die Objekte 2, 6, 7 Transitionen, die Objekte 4, 5 Aktionsblöcke, das Objekt 1a, 1b eine ODER-Verzweigung und das Objekt 3a, 3b eine UND-Verzweigung darstellen. Innerhalb der Funktionsplangrafik ist ein Cursor 8 angezeigt, welcher durch Betätigen der Cursortasten "Pfeil links" (←), "Pfeil rechts" (→) "Pfeil oben"

(↑) und "Pfeil unten" (↓) einer hier nicht dargestellten Tastatur des Programmiergerätes horizontal und vertikal, wie im folgenden noch gezeigt wird, objektweise verschiebbar ist.

Es wird zunächst angenommen, daß der Cursor 8 vor der Transition 2 positioniert ist (Fig. 1a) und daß ein Programmierer die Cursortaste "Pfeil unten" (↓) betätigt. Dies bewirkt, daß ein Cursor-Treiberprogramm, das Teil eines Software-Werkzeugs zum Erstellen der Funktionsplangrafik sein kann, den Cursor 8 in die Lücke zwischen der Transition 2 und dem oberen Rahmen 3a der UND-Verzweigung 3a, 3b positioniert (Fig. 1b), wobei der Cursor 8 horizontal dargestellt ist, da die Objekte 2, 3a, 3b, 6 vertikal angeordnet sind. Drückt der Programmierer wiederum die Cursortaste "Pfeil unten" (↓), so positioniert das Software-Werkzeug den Cursor 8 in die Lücke zwischen dem unteren Rahmen 3b der UND-Verzweigung 3a, 3b und der Transition 6 (Fig. 1c).

Zur Verdeutlichung einer objektweisen horizontalen Cursorpositionierung (Cursorverschiebung) wird auf Fig. 2 verwiesen. Die in den Fig. 1 und 2 vorkommenden gleichen Teile sind mit gleichen Bezugszeichen versehen. Es ist angenommen, daß der Cursor 8 oberhalb der Transition 6 positioniert ist (Fig. 2a) und daß der Cursor 8 innerhalb der UND-Verzweigung 3a, 3b positioniert werden soll. Wie beschrieben, erfolgt die Verschiebung des Cursors 8 objektweise, und es ist daher erforderlich, neben der Cursortaste "Pfeil nach oben" (↑) eine weitere Taste zu drücken, damit der Cursor 8 zwischen den Rahmen 3a, 3b der UND-Verzweigung 3a, 3b positionierbar ist. Betätigt der Programmierer die Cursortaste "Pfeil oben" (↑) und eine weitere Taste, z. B. die Shift-Taste, so positioniert das Software-Werkzeug den Cursor 8 innerhalb der UND-Verzweigung 3a, 3b links neben dem Aktionsblock 4 (Fig. 2b), wobei der Cursor 8 vertikal dargestellt wird, da innerhalb der UND-Verzweigung 3a, 3b nur horizontal angeordnete Objekte 4, 5 positioniert sind. Innerhalb dieser UND-Verzweigung 3a, 3b erfolgt eine Cursorverschiebung wiederum objektweise, z. B. bewirkt das Drücken der Cursortaste "Pfeil rechts" (→) die Verschiebung des Cursors 8 in die Lücke zwischen dem Aktionsblock 4 und 5 (Fig. 2c). Für den Fall, daß wiederum Objekte außerhalb der UND-Verzweigung 3a, 3b auszuwählen sind, ist wiederum die Shift-Taste sowie eine der Cursortasten "Pfeil links" (←), "Pfeil rechts" (→), "Pfeil oben" (↑) und "Pfeil unten" (↓) zu drücken.

Zur Verdeutlichung einer Einfüge-Operation wird auf Fig. 3 verwiesen. Die in den Fig. 1 bis 3 vorkommenden gleichen Teile sind wiederum mit gleichen Bezugszeichen versehen. Es ist angenommen, daß der Cursor 8 innerhalb der UND-Verzweigung 3a, 3b in der Lücke zwischen dem Aktionsblock 4 und 5 positioniert ist (Fig. 3a). Nach der Aktion "Schritt einfügen", welche z. B. durch Drücken einer Funktionstaste, z. B. der Funktionstaste F1 oder F2, ermöglicht wird, fügt das Software-Werkzeug zwischen den Aktionsblöcken 4 und 5 einen weiteren Aktionsblock 9 ein und positioniert nun den Cursor 8 zwischen diesem Aktionsblock 9 und dem Aktionsblock 5 (Fig. 3b).

Ein Einfügen eines Objektes innerhalb vertikal angeordneter Objekte erfolgt in der gleichen Art und Weise. Es ist angenommen, daß der Cursor 8 in der Lücke zwischen einem Aktionsblock 10 und einer Transition 11 positioniert ist (Fig. 3c). Nach der Aktion "Aktionsblock einfügen" (Drücken einer Funktionstaste) fügt das Software-Werkzeug einen weiteren Aktionsblock 12 an der Cursorposition ein, wobei das Software-Werkzeug den

Cursor 8 in vertikaler Richtung verschiebt (Fig. 3d) und den Cursor 8 zwischen den Aktionsblöcken 10 und 12 positioniert.

Vorteilhaft kann nach einer Cursorpositionierung durch mehrmaliges Betätigen einer Taste (ohne Maus und ohne Cursortasten) eine ganze Folge von Objekten eingegeben werden. Ein Navigieren in großen Diagrammen ist einfacher, weil auch komplexe Objekte mit einer Taste übersprungen werden. Mehrfachselektionen in großen Diagrammen werden ebenso wie das Navigieren vereinfacht. Ein umständliches Suchen der nächsten Einfüge-Position entfällt, weil der Cursor mit positioniert wird. Darüber hinaus ist die Einfüge-Position eindeutig erkennbar.

Selbstverständlich ist es möglich, den Cursor beim Betätigen der Cursortasten nicht in die Lücke zwischen zwei Objekten zu positionieren, sondern daß beim Betätigen der Cursortasten der Cursor von einem Objekt zum anderen springt. Nach welcher Art und Weise der Cursor positioniert (verschoben) werden soll, ist parametrierbar. Man kann in der Weise vorgehen, daß vor der Erstellung einer Funktionsplangrafik das Software-Werkzeug menügesteuert den Programmierer auffordert, die entsprechenden Parameter vorzugeben. Für den Fall, daß der Cursor von einem Objekt zum nächsten springen soll, ist ferner parametrierbar, ob ein einzufügendes Objekt vor oder hinter dem Cursor einzufügen ist.

#### Patentansprüche

1. Programmiergerät, welches eine Tastatur und eine Anzeigeeinheit sowie ein Software-Werkzeug zum Erstellen eines Steuerprogramms für ein Automatisierungsgerät in Form einer Grafik umfaßt, die mit auf der Anzeigeeinheit darstellbaren Objekten (1 ... 7, 9 ... 12) versehen ist, wobei die Objekte (1 ... 7, 9 ... 12) horizontal und vertikal oder in Form einer Netzstruktur angeordnet sind und zueinander in Beziehung stehen, dadurch gekennzeichnet, daß in der Grafik ein Cursor (8) anzeigbar ist und durch Betätigen einer Cursortaste ("Pfeil links" (←), "Pfeil rechts" (→), "Pfeil oben" (↑), "Pfeil unten" (↓)) die Cursorpositionierung objektweise erfolgt.
2. Programmiergerät nach Anspruch 1, dadurch gekennzeichnet, daß die Grafik eine Funktionsplangrafik ist, die mit Objekten (1 ... 7, 9 ... 12) in Form von Transitionen (2, 6, 7, 11), UND- und/oder ODER-Verzweigungen (1a, 1b, 3a, 3b) und Aktionsblöcken (4, 5, 9, 10, 12) versehen ist.
3. Programmiergerät nach Anspruch 1, dadurch gekennzeichnet, daß die Grafik ein Nassi-Shneiderman-Diagramm oder ein Petri-Netz ist.
4. Programmiergerät nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß durch Betätigen einer Cursortaste der Cursor in die Lücke zwischen zwei Objekten springt.
5. Programmiergerät nach Anspruch 4, dadurch gekennzeichnet, daß an der Cursorposition ein Objekt einfügbar ist, wobei automatisch Platz zum Einfügen des Objektes geschaffen wird.
6. Programmiergerät nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß durch Betätigen einer Cursortaste der Cursor von einem Objekt zum nächsten springt, wobei in einem Einfüge-Modus ein einzufügendes Objekt nach Maßgabe einer in einem Speicher des Programmiergerätes hinter-

legten Bearbeitungsvorschrift vor oder hinter dem Cursor positioniert wird.

7. Programmiergerät nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß Objekte in Gruppen zusammenfaßbar sind und die Cursorpositionierung objektgruppenweise erfolgt.

8. Programmiergerät nach Anspruch 7, dadurch gekennzeichnet, daß das Betätigen einer Cursortaste in Verbindung mit der Betätigung einer weiteren Taste eine Cursorpositionierung innerhalb einer Objektgruppe objektweise bewirkt.

9. Programmiergerät nach Anspruch 7, dadurch gekennzeichnet,

- daß die Cursorpositionierung innerhalb einer Gruppe durch Betätigen einer Cursortaste objektweise erfolgt,
- daß, für den Fall, daß der Cursor innerhalb einer Objektgruppe positioniert ist, eine Betätigung einer Cursortaste in Verbindung mit der Betätigung einer weiteren Taste bewirkt, daß der Cursor außerhalb der Objektgruppe positioniert wird, und
- daß die Cursorpositionierung außerhalb einer Gruppe durch Betätigen einer Cursortaste objektgruppenweise erfolgt.

10. Programmiergerät nach einem der Ansprüche 1 bis 9, dadurch gekennzeichnet, daß der Cursor (8) horizontal und/oder vertikal darstellbar ist.

Hierzu 3 Seite(n) Zeichnungen

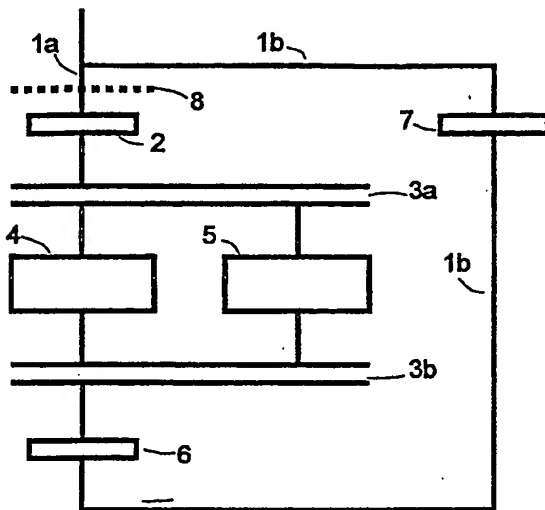


FIG 1a

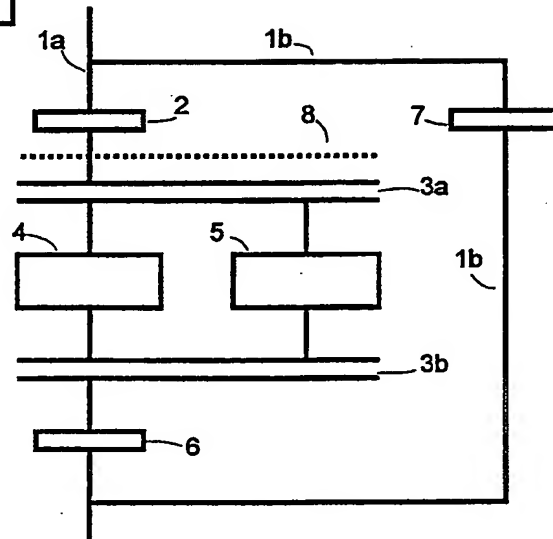


FIG 1b

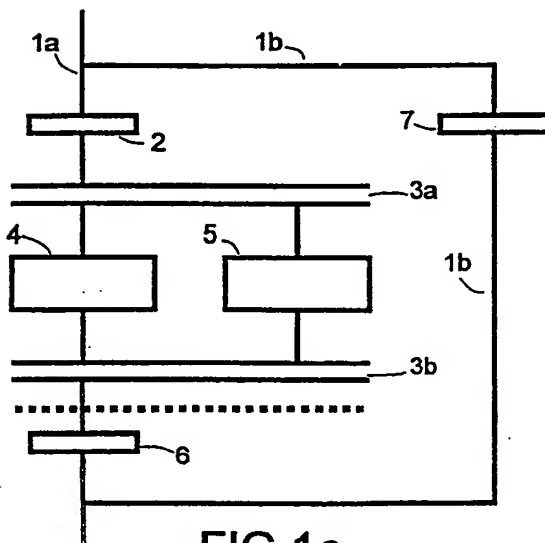


FIG 1c

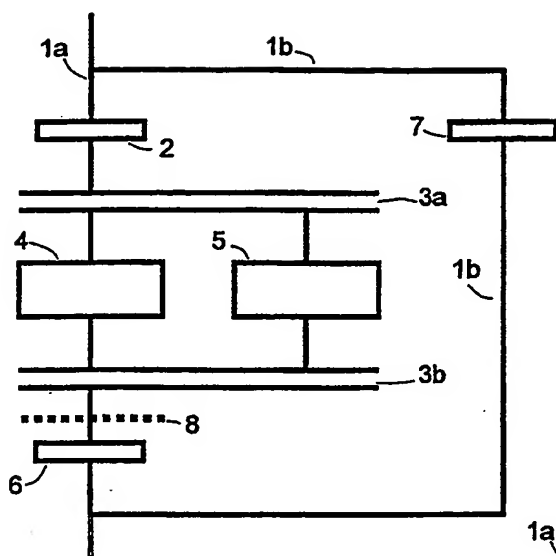


FIG 2a

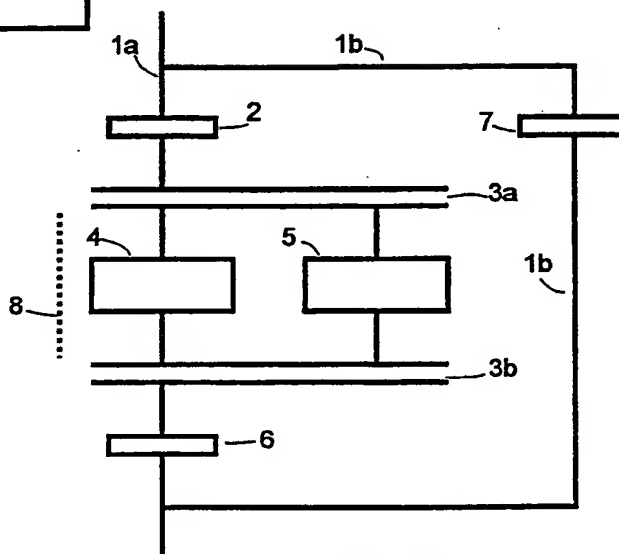


FIG 2b

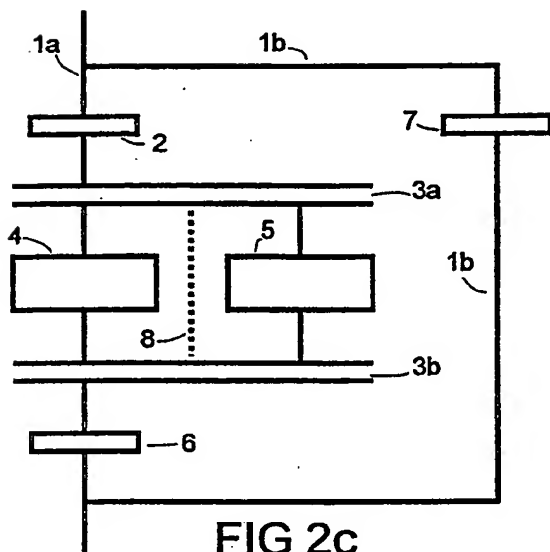


FIG 2c

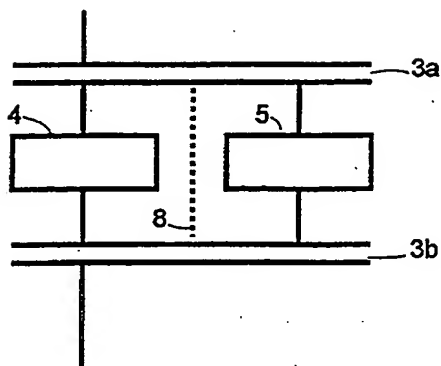


FIG 3a

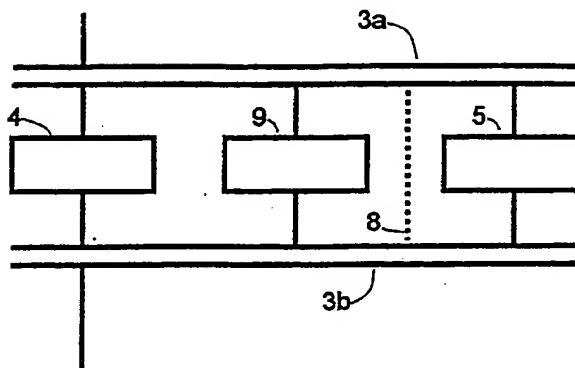


FIG 3b

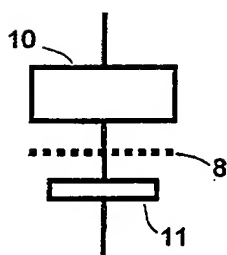


FIG 3c

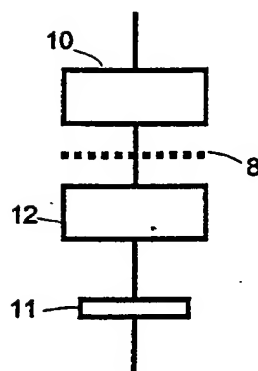


FIG 3d